**Chapter 05. 웹 서버 만들기**

* 요청 라우팅 하기

앞서 학습한 use( )를 통한 미들웨어 사용은 요청 url이 무엇인지 일일이 확인해야 하므로, 이 문제를 해결하기 위해 router 미들웨어를 사용한다.

1. 라우터 객체 생성 🡪 2. 라우터 객체에 요청 패스 별 라우팅 함수 등록 🡪 3. 라우터 객체를 미들웨어로 등록

router.route(요청패스).get(함수)  
router.route(요청패스).post(함수)

[public/login2.html]

<!-- 요청 패스를 action 사용해서 추가함 -->

<form method="post" action = "/process/login">

[app8.js]

// 라우터 객체 참조

var router = express.Router();

// 라우팅 함수 등록

router.route('/process/login').post(function(req, res){

console.log('/process/login 처리함');

var paramId = req.body.id || req.query.id;

var paramPassword = req.body.password || req.query.password;

res.writeHead('200', {'Content-Type': 'text/html;charset=utf8'});

res.write('<h1>Express 서버에서 응답한 결과입니다.</h1>');

res.write('<div><p>Param id : ' + paramId + '</p></div>');

res.write('<div><p>Param password : ' + paramPassword + '</p></div>');

res.write("<br><br><a href='/public/login2.html'>로그인 페이지로 돌아가기</a>");

res.end();

});

// 라우터 객체를 app 객체에 미들웨어로 등록

app.use('/', router);

* Login2.html에서 요청프로세스 /process/login으로 넘어온 정보를 라우팅 함수에서 처리해 응답 페이지를 전송하는 과정
* URL 파라미터를 사용하기

URL 파라미터는 요청 파라미터와 달리 URL 주소의 일부로 들어간다. URL 파라미터를 전달받아 처리하는 방식 확인!

[app8\_02.js]

라우팅 함수에 ‘/process/login/:name’ 으로 수정할 경우, login 뒤의 값을 파라미터로 처리하겠다는 의미. 이렇게 지정된 파라미터는 **req.params** 객체 안에 저장되어 전달된다.

* 이 경우, :name에 넣어 전달된 값은 **req.params.name** 속성으로 접근할 수 있으며 이를 **토큰(Token)**이라 부른다.

// 라우팅 함수 등록

router.route('/process/login/:name').post(function(req, res){

console.log('/process/login/:name 처리함');

var paramName = req.params.name;

var paramId = req.body.id || req.query.id;

var paramPassword = req.body.password || req.query.password;

res.writeHead('200', {'Content-Type': 'text/html;charset=utf8'});

res.write('<h1>Express 서버에서 응답한 결과입니다.</h1>');

res.write('<div><p>Param name : ' + paramName + '</p></div>');

res.write('<div><p>Param id : ' + paramId + '</p></div>');

res.write('<div><p>Param password : ' + paramPassword + '</p></div>');

res.write("<br><br><a href='/public/login2.html'>로그인 페이지로 돌아가기</a>");

res.end();

});

[/public/login3.html]

<form method="post" action = "/process/login/izone">

* 오류 페이지 보여주기

지정된 요청 패스 이외의 요청이 들어왔을 때 오류페이지 설정 방법

[app8.js] 라우팅 함수 하단에 추가

// 등록되지 않은 패스에 대해 페이지 오류 응답

app.all('\*', function(req, res){

res.status(404).send('<h1>ERROR~페이지를 찾을 수 없습니다.</h1>');

});

* Express-error-handler 미들웨어로 오류 페이지 보내기

[app8.js] 추가

// 모든 router 처리가 끝난 후 404 오류 페이지 처리

var errorHandler = expressErrorHandler({

static: {

'404': './public/404.html'

}

});

app.use(expressErrorHandler.httpError(404));

app.use(errorHandler);

* 토큰과 함께 요청한 정보 처리하기

요청 URL에 포함된 URL 파라미터 사용방법 재 학습. get( ) 타입으로 전달된다

[app10.js]

// 라우팅 함수 등록

router.route('/process/users/:id').get(function(req, res) {

    console.log('/process/users/:id 처리함.');

// URL 파라미터 확인

    var paramId = req.params.id;

    console.log('/process/users와 토큰 %s를 이용해 처리함.', paramId);

    res.writeHead('200', {'Content-Type':'text/html;charset=utf8'});

    res.write('<h1>Express 서버에서 응답한 결과입니다.</h1>');

    res.write('<div><p>Param id : ' + paramId + '</p></div>');

    res.end();

});

* 브라우저에서 localhost:3000/process/users/2 이런 식으로 입력